Wingtip Vortex Preservation Using a Coupled Vortex Particle Method and Computational Fluid Dynamics Solver

S. J. Huntley*, D. Jones† and A. Gaitonde†

Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, United Kingdom

This paper presents a coupled Vortex Particle Method-Computational Fluid Dynamics solver. The Vortex Particle Method solution is used to correct the Computational Fluid Dynamics. This is achieved by using the Split Velocity Method, which includes the influence of the particles through their induced velocities and additional source terms. The coupled solver is demonstrated on a two-dimensional and three-dimensional test case and the results are compared to the solutions of the CFD solver on its own using a coarse mesh and a fine mesh. It is shown that the hybrid solver preserves the tip vortex on a coarse mesh and is computationally more efficient than using the fine mesh.

Nomenclature

$\hat{\hat{u}}, \hat{\hat{v}}, \hat{\hat{w}}$	CFD particle-induced velocity components
$\hat{\hat{u}}_p, \hat{\hat{v}}_p, \hat{\hat{w}}_p$	VPM particle-induced velocity components
Γ	circulation
γ	ratio of specific heats
μ	dynamic viscosity
ω	vorticity
ρ	density
$\tilde{u}, \tilde{v}, \tilde{w}$	background velocity components
E	energy
p	pressure
Pr	Prandtl number
Re	Reynolds number
t	time
u, v, w	total velocity components

I. Introduction

Understanding the effect of concentrated vortices interacting with a lifting surface is important as it influences the aerodynamics amongst other things. Examples of this include helicopter blade-vortex interaction and aircraft take-off and landing. Blade-vortex interaction occurs when the trailing wake from a helicopter rotor interacts with the oncoming blade. This causes unsteady loading on the blades and aerodynamic noise.[1] Wingtip vortices that are generated on aircraft take-off and landing determine separation distances to ensure the safety of following aircraft. The ability to accurately predict these types of flows represents a challenge in fluid simulation methods. The literature shows that Lagrangian, Eulerian and hybrid Eulerian/Lagrangian Computational Fluid Dynamics (CFD) techniques have been applied to these kinds of flows.

Eulerian CFD methods are widely used for aerospace applications due to their ability to accurately and efficiently resolve the flow near solid boundaries. However, they suffer from diffusive behaviour meaning

*Research Associate, AIAA member

†Senior Aerodynamics Lecturer, AIAA member

their application to the simulation of flows with highly concentrated vortical regions has been limited, as accurately resolving these flow features would require a very fine mesh throughout the computational domain and result in a very large computational overhead. Eulerian CFD methods used to compute vortical flows range in complexity and computational cost from Reynolds-Averaged Navier-Stokes (RANS) codes to Direct Numerical Simulation (DNS) methods. The RANS approach was used in Potsdam *et al.*[2] to calculate helicopter blade-vortex interactions. It is the least complex of the CFD methods but can become computationally expensive as it requires a very fine mesh to prevent excessive diffusion. This is highlighted in the study by Abate,[3] where RANS was used to study an airfoil-vortex interaction in two dimensions however, the extension of this to three dimensions is unlikely due to the cost of using a fine mesh. DNS is the most accurate of the CFD methods as it resolves all the turbulent length scales and does not model anything however, it is computationally very expensive and so far has only been applied to the simplest of flows such as a stalled NACA 0012 aerofoil[4] due to its cost. Other studies have utilised high-order algorithms to increase the accuracy of the CFD in high gradient regions such as used by Svard *et al.* to study the interaction of a vortex with a NACA 0012 airfoil.[5] An alternative that has been explored is to adopt a chimera grid approach that uses overlapped grids to transport the vortex on a locally fine mesh across a coarser background mesh as investigated by Wolf.[6]

Lagrangian methods benefit from the fact that they do not suffer from any diffusive effects and computational effort can be reduced as computational elements only need to be placed in regions of interest. However these elements are isotropic in nature causing difficulty when modelling solid boundaries.[7] Lagrangian methods have repeatedly been used over the past few decades to model aircraft wakes and tip vortices. Rossow[8] used a vortex filament method to model a simplified wake structure whilst Smith and Kroo used a vortex panel method for the same purpose.[9] More recently, Chatelain *et al.* used a vortex particle method with a billion particles to model the aircraft wake.[10] Vortex Lattice methods are the most widely used method for rotor wake prediction to represent the shed and trailed vorticity generated by the rotor, see for example Röttgermann *et al.*[11] and the study by Padakannaya.[12] Vortex Particle Methods have also been used to predict the rotor wake and blade-vortex interactions.

More recently a number of studies have looked at hybrid Eulerian/Lagrangian solvers to compute vortical flows. These studies couple a RANS CFD solver to a particle-based Lagrangian method and employ a domain decomposition approach so that the CFD is only applied in the near-body region and the particle method is used in the outer region. Most methods also employ an 'overlap' region to couple the two methods where both solvers are used and the solutions are interpolated between each other. The first example of this is by Sitaraman[13] who coupled a RANS solver to a Particle Vorticity Transport Method (PVTM) to simulate rotor wakes. Anusonti-Inthra and Floros[14] extend this to a viscous PVTM to model the flow in isolated wing wakes. Zhao *et al.* employ a viscous vortex particle method together with two different RANS solvers to investigate rotor wake flow.[15] Stone *et al.*[16,17] used an overset Unsteady RANS flow solver coupled to a vortex particle method to investigate rotor blade-vortex interactions. Pahla *et al.*[18] take a slightly different approach, whilst the domain is still decomposed into regions the Lagrangian method is applied to the entire domain whilst the Eulerian solver is only applied to the region close to the solid boundary. Essentially, they use the CFD to correct the particle method in the near-field.

In this work, we propose a new, hybrid method that couples a RANS solver with a Vortex Particle Method. The domain is not decomposed into regions; instead the VPM is used to correct the CFD. This is done by including the influence of the particles in the CFD solution through the Split Velocity Method, which has previously been used in the simulation of gusts.[19,20]

II. Methodology

The Navier-Stokes equations govern the motion of fluid flow. The equations are most commonly described using the Eulerian formulation. This is the approach used here within the CFD solver DLR-TAU, where the flow quantities are considered functions of the spatial location as they change with time. For vortex transit this requires fine meshes throughout the computational domain leading to very high computational cost. However, the Split Velocity Method recently developed and implemented for gusts can be extended to produce an efficient method for vortex transit if a suitable alternative Lagrangian model of the vortex is available. The Vortex Particle Method solves the Navier-Stokes equations in velocity-vorticity form by considering the problem as a collection of fluid elements. The Split Velocity Method allows the vortex transit to be modelled by the Lagrangian scheme and the CFD can then compute the flow field minus the model

prediction. The key advantage is that coarser meshes can be used without the vortex being dissipated. This section first describes the CFD formulation and then presents the VPM. Finally, the coupling method is explained along with techniques used to speed up the computation.

II.A. CFD solver

The unsteady Navier-Stokes equations are given by

$$\begin{aligned}
&\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ u(\rho E+p)\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ v(\rho E+p)\end{bmatrix}\\
&+\frac{\partial}{\partial z}\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ w(\rho E+p)\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ \sigma_{xz}\\ u\sigma_{xx}+v\sigma_{xy}+w\sigma_{xz}+q_x\end{bmatrix}\\
&+\frac{\partial}{\partial y}\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ \sigma_{zy}\\ u\sigma_{xy}+v\sigma_{yy}+w\sigma_{yz}+q_y\end{bmatrix}+\frac{\partial}{\partial z}\begin{bmatrix}0\\ \sigma_{xz}\\ \sigma_{yz}\\ \sigma_{zz}\\ u\sigma_{xz}+v\sigma_{yz}+w\sigma_{zz}+q_z\end{bmatrix}=0
\end{aligned} \tag{1}$$

where the energy and pressure are given by

$$E=\frac{p}{\rho(\gamma-1)}+\frac{1}{2}(u^2+v^2+w^2) \tag{2}$$

and

$$p=(\gamma-1)(\rho E-\frac{\rho}{2}(u^2+v^2+w^2)) \tag{3}$$

respectively.

In this work the Navier-Stokes equations, Eq. (1), are solved using a cell-centred finite volume scheme on an unstructured mesh. The convective flux of the mean flow equations is approximated using a central difference scheme with matrix dissipation and the equations are integrated in time using a dual-timestepping method, where the time-derivative is discretised using a second-order Backward Difference Formula (BDF). Each time step is converged using the Backward Euler method in which the linear system is solved using the Lower-Upper Symmetric Gauss-Seidel (LUSGS) scheme. Although DLR-TAU employs a multigrid acceleration technique, it has been switched off for this work.

II.B. Vortex Particle Method

In a Vortex Particle Method, the flow is discretised into particles with concentrated circulations or strengths, which are then convected according to the Lagrangian form of the transport equation. Rather than using point vortices which have a singularity, vortex blobs or regularized vortex particles are used, which are particles with a finite core size. In this case, the vorticity field is given by

$$\omega(\mathbf{x},t)=\sum_{i=1}^{N}\mathbf{\Gamma}_i(t)\zeta_{\sigma_i}(\mathbf{x}-\mathbf{x}_i(t)) \tag{4}$$

where ω is the vorticity, $\mathbf{\Gamma}_i=[\Gamma_{x_i},\Gamma_{y_i},\Gamma_{z_i}]$ is the strength vector of particle i, ζ_σ is the regularized smoothing kernel[21] and σ_i is the radius of particle i. In this work, all particles have the same radius so $\sigma_i=\sigma$.

For a three-dimensional case, the regularized smoothing kernel is given by

$$\zeta_\sigma=\frac{1}{\sigma^3}\zeta\left(\frac{|\mathbf{x}|}{\sigma}\right) \tag{5}$$

where $\zeta(\rho)$ is the smoothing or cutoff function. The cutoff function should be smooth and accurate[7] and several choices are listed in the paper by Wincklemans and Leonard.[22] In this work the second-order 3D gaussian smoothing, given by Eq. (6), is used.

$$\zeta\left(\rho\right) = \left(\frac{2}{\pi}\right)^{1/2} \exp\left(-\frac{\rho^2}{2}\right) \tag{6}$$

The equations of motion for a regularized vortex particle method are given by

$$\frac{d}{dt}\mathbf{x}\left(t\right) = \mathbf{u}_\sigma\left(\mathbf{x}\left(t\right), t\right) \tag{7}$$

$$\frac{d}{dt}\mathbf{\Gamma}\left(t\right) = \left(\mathbf{\Gamma}\left(t\right)\cdot\nabla\right)\mathbf{u}_\sigma\left(\mathbf{x}\left(t\right), t\right) + \left(\nu\nabla^2\omega\right)V \tag{8}$$

Equation (8) consists of a viscous diffusion part,

$$\frac{d}{dt}\mathbf{\Gamma}\left(t\right) = \left(\nu\nabla^2\omega\right)V, \tag{9}$$

where V is the volume asscoiated with a single particle, and a stretching component,

$$\frac{d}{dt}\mathbf{\Gamma}\left(t\right) = \left(\mathbf{\Gamma}\left(t\right)\cdot\nabla\right)\mathbf{u}_\sigma\left(\mathbf{x}\left(t\right), t\right). \tag{10}$$

Formulating the stretching term as given in Eq. (10) results in the classical scheme.[22] However, this equation can also be written differently leading to either the transpose scheme or the mixed scheme.[22] All schemes would be identical if $\omega = \nabla\times\mathbf{u}$ at all times but due to divergence this is not the case. The transpose scheme is favoured as it leads to exact conservation of total vorticity[22] however, the classical scheme has also been shown to perform well.[21]

The velocity $\mathbf{u}_\sigma\left(\mathbf{x}, t\right)$ is computed using the generalized Biot-Savart equation, which is

$$\mathbf{u}_\sigma\left(\mathbf{x}, t\right) = \sum_{i=1}^{N}\mathbf{K}_\sigma\left(\mathbf{x}-\mathbf{x}_i(t)\right)\times\mathbf{\Gamma}_i(t) \tag{11}$$

which when written in terms of Green's function for the Laplace equation is given by

$$\mathbf{u}_\sigma\left(\mathbf{x}, t\right) = -\sum_{i=1}^{N}\nabla G(\mathbf{x}-\mathbf{x}_i(t))\times g_\sigma\left(\frac{\mathbf{x}}{\sigma}\right)\mathbf{\Gamma}_i(t) \tag{12}$$

where $g_\sigma(\rho)$ is given by

$$g_\sigma(\rho) = erf\left(\frac{\rho}{2^{1/2}}\right) - \left(\frac{2}{\pi}\right)^{1/2}\rho\exp(-\rho^2/2) \tag{13}$$

From this point, the subscript σ is dropped from the velocity notation. The Biot-Savart equation, Eq. (12), is solved twice once for the VPM particle-induced velocities which will be denoted $\hat{\mathbf{u}}_p = [\hat{u}_p, \hat{v}_p, \hat{w}_p]$ and once for the CFD particle-induced velocities, denoted by $\hat{\mathbf{u}} = [\hat{u}, \hat{v}, \hat{w}]$. The convection equation, Eq. (7), is solved using a fourth order Runge-Kutta method. The stretching component of Eq. (8) and the velocity equation, Eq. (12), are solved using a Fast Multipole Method (FMM), which is described in section II.C.2.

II.C. Coupling method

The CFD solver and the VPM are coupled together using the process shown in Figure 1, which shows that the whole solution process involves many steps. The main steps in the coupling involve seeding the particles from the CFD vorticity and computing the particle induced velocities and source terms for the Split Velocity Method. This is described first, followed by a fast summation technique, which is used to reduce the computational cost of the process. This is necessary as the computation of the velocities and source terms requires summation over a large number of elements. Finally, the particle seeding routine is explained.

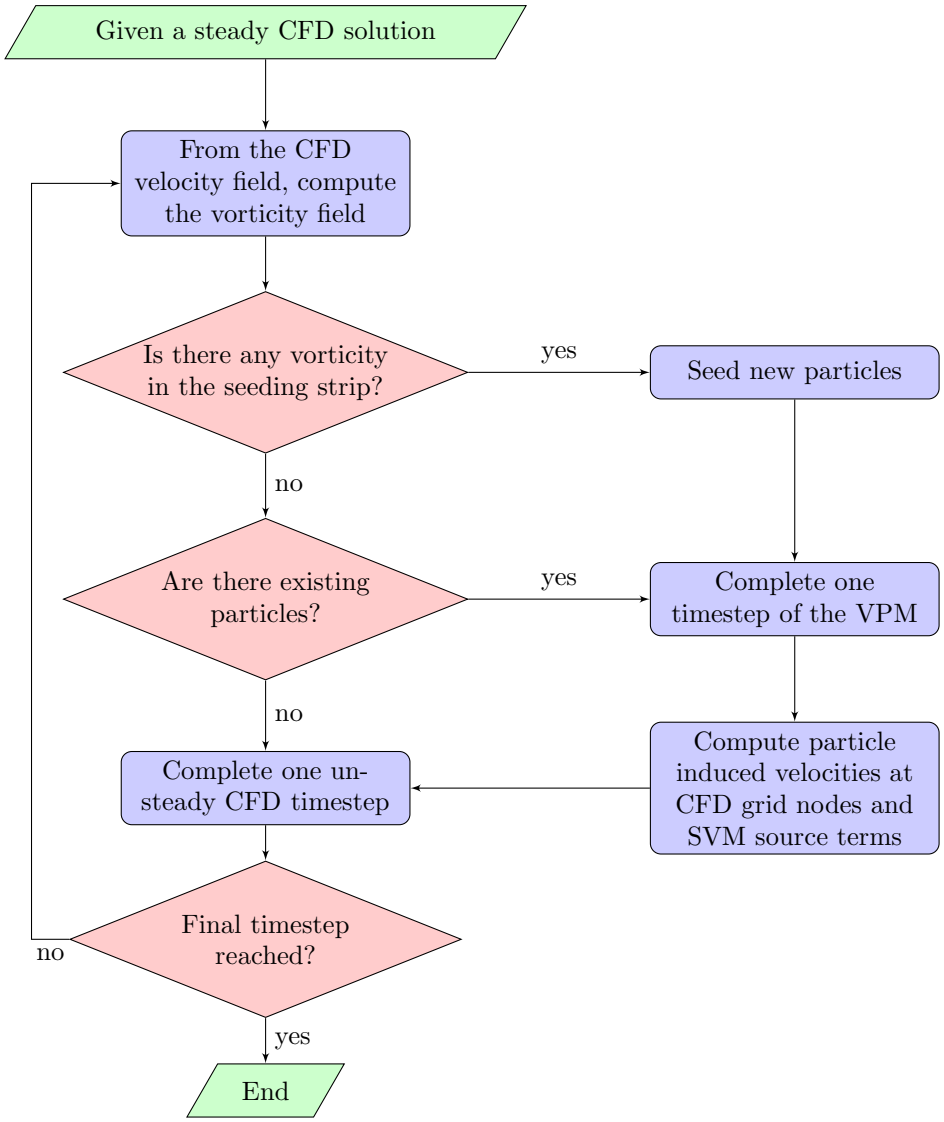


Figure 1. Flowchart of the coupled CFD-VPM solver

II.C.1. Split Velocity Method

The formulation for the Split Velocity Method begins with the unsteady Navier-Stokes equations on a fixed mesh, Eq. (1). Then the velocity and energy are decomposed as

$$u = \tilde{u} + \hat{\hat{u}} \qquad v = \tilde{v} + \hat{\hat{v}} \qquad w = \tilde{w} + \hat{\hat{w}} \qquad E = \tilde{E} + \hat{\tilde{E}} + \hat{\hat{E}} \tag{14}$$

where $\hat{\hat{u}}$, $\hat{\hat{v}}$ and $\hat{\hat{w}}$ are the induced vortex particle velocity components as explained above and $\hat{\hat{E}}$ is computed by substituting the velocity decompositions into Eq. (2), which after some manipulation gives

$$E = \underbrace{\frac{p}{\rho(\gamma-1)} + \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2)}_{\tilde{E}} + \underbrace{(\tilde{u}\hat{\hat{u}} + \tilde{v}\hat{\hat{v}} + \tilde{w}\hat{\hat{w}})}_{\hat{\tilde{E}}} + \underbrace{\frac{1}{2}(\hat{\hat{u}}^2 + \hat{\hat{v}}^2 + \hat{\hat{w}}^2)}_{\hat{\hat{E}}} \tag{15}$$

The pressure remains unchanged and is therefore given by Eq. (3). The Navier-Stokes equations for the Split Velocity Method are then obtained by substituting the decompositions in Eq. (14) into the unsteady

Navier-stokes equations given by Eq. (1). This gives

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho(\tilde{u}+\hat{u})\\ \rho(\tilde{v}+\hat{v})\\ \rho(\tilde{w}+\hat{w})\\ \rho(\tilde{E}+\hat{\tilde{E}}+\hat{E})\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho(\tilde{u}+\hat{u})\\ \rho(\tilde{u}+\hat{u})^2+p\\ \rho(\tilde{u}+\hat{u})(\tilde{v}+\hat{v})\\ \rho(\tilde{u}+\hat{u})(\tilde{w}+\hat{w})\\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{E}))(\tilde{u}+\hat{u})\end{bmatrix}$$
$$+\frac{\partial}{\partial y}\begin{bmatrix}\rho(\tilde{v}+\hat{v})\\ \rho(\tilde{v}+\hat{v})(\tilde{u}+\hat{u})\\ \rho(\tilde{v}+\hat{v})^2+p\\ \rho(\tilde{v}+\hat{v})(\tilde{w}+\hat{w})\\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{E}))(\tilde{v}+\hat{v})\end{bmatrix}+\frac{\partial}{\partial z}\begin{bmatrix}\rho(\tilde{w}+\hat{w})\\ \rho(\tilde{w}+\hat{w})(\tilde{u}+\hat{u})\\ \rho(\tilde{w}+\hat{w})(\tilde{v}+\hat{v})\\ \rho(\tilde{w}+\hat{w})^2+p\\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{E}))(\tilde{w}+\hat{w})\end{bmatrix}$$
$$+\frac{\partial}{\partial x}\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ \sigma_{xz}\\ (\tilde{u}+\hat{u})\sigma_{xx}+(\tilde{v}+\hat{v})\sigma_{xy}+(\tilde{w}+\hat{w})\sigma_{xz}+q_x\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ \sigma_{yz}\\ (\tilde{u}+\hat{u})\sigma_{xy}+(\tilde{v}+\hat{v})\sigma_{yy}+(\tilde{w}+\hat{w})\sigma_{yz}+q_y\end{bmatrix}$$
$$+\frac{\partial}{\partial z}\begin{bmatrix}0\\ \sigma_{xz}\\ \sigma_{yz}\\ \sigma_{zz}\\ (\tilde{u}+\hat{u})\sigma_{xz}+(\tilde{v}+\hat{v})\sigma_{yz}+(\tilde{w}+\hat{w})\sigma_{zz}+q_z\end{bmatrix}=0 \tag{16}$$

where,

$$\sigma_{xx}=\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{u}}{\partial x}-\frac{\partial\tilde{v}}{\partial y}-\frac{\partial\tilde{w}}{\partial z}\right)+\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{u}}{\partial x}-\frac{\partial\hat{v}}{\partial y}-\frac{\partial\hat{w}}{\partial z}\right)\quad \sigma_{yy}=\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{v}}{\partial y}-\frac{\partial\tilde{u}}{\partial x}-\frac{\partial\tilde{w}}{\partial z}\right)+\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{v}}{\partial y}-\frac{\partial\hat{u}}{\partial x}-\frac{\partial\hat{w}}{\partial z}\right)$$

$$\sigma_{zz}=\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\tilde{w}}{\partial z}-\frac{\partial\tilde{v}}{\partial y}-\frac{\partial\tilde{u}}{\partial x}\right)+\frac{2}{3}\frac{\mu}{Re}\left(2\frac{\partial\hat{w}}{\partial z}-\frac{\partial\hat{v}}{\partial y}-\frac{\partial\hat{u}}{\partial x}\right)\quad \sigma_{xy}=\frac{\mu}{Re}\left(\frac{\partial\tilde{u}}{\partial y}+\frac{\partial\tilde{v}}{\partial x}\right)+\frac{\mu}{Re}\left(\frac{\partial\hat{u}}{\partial y}+\frac{\partial\hat{v}}{\partial x}\right)$$

$$\sigma_{xz}=\frac{\mu}{Re}\left(\frac{\partial\tilde{u}}{\partial z}+\frac{\partial\tilde{w}}{\partial x}\right)+\frac{\mu}{Re}\left(\frac{\partial\hat{u}}{\partial z}+\frac{\partial\hat{w}}{\partial x}\right)\quad \sigma_{yz}=\frac{\mu}{Re}\left(\frac{\partial\tilde{v}}{\partial z}+\frac{\partial\tilde{w}}{\partial y}\right)+\frac{\mu}{Re}\left(\frac{\partial\hat{v}}{\partial z}+\frac{\partial\hat{w}}{\partial y}\right)$$

$$q_x=-\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial x}\quad q_y=-\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial y}$$

$$q_z=-\frac{\mu}{Pr}\frac{1}{Re}\frac{1}{(\gamma-1)M_\infty^2}\frac{\partial T}{\partial z}\quad T=\frac{\gamma M_\infty^2 p}{\rho}$$

where, Pr, μ and Re are the Prandtl number, dynamic viscosity and Reynolds number respectively. Separating the induced velocities from the rest of the solution and after some manipulation of the terms, the

Navier-Stokes equations are given as

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho\tilde{u}\\ \rho\tilde{v}\\ \rho\tilde{w}\\ \rho\tilde{E}\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho(\tilde{u}+\hat{\tilde{u}})\\ \rho\tilde{u}(\tilde{u}+\hat{\tilde{u}})+p\\ \rho\tilde{v}(\tilde{u}+\hat{\tilde{u}})\\ \rho\tilde{w}(\tilde{u}+\hat{\tilde{u}})\\ \rho\tilde{E}(\tilde{u}+\hat{\tilde{u}})+p\tilde{u}\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho(\tilde{v}+\hat{\tilde{v}})\\ \rho\tilde{u}(\tilde{v}+\hat{\tilde{v}})\\ \rho\tilde{v}(\tilde{v}+\hat{\tilde{v}})+p\\ \rho\tilde{w}(\tilde{v}+\hat{\tilde{v}})\\ \rho\tilde{E}(\tilde{v}+\hat{\tilde{v}})+p\tilde{v}\end{bmatrix}+\frac{\partial}{\partial z}\begin{bmatrix}\rho(\tilde{w}+\hat{\tilde{w}})\\ \rho\tilde{u}(\tilde{w}+\hat{\tilde{w}})\\ \rho\tilde{v}(\tilde{w}+\hat{\tilde{w}})\\ \rho\tilde{w}(\tilde{w}+\hat{\tilde{w}})+p\\ \rho\tilde{E}(\tilde{w}+\hat{\tilde{w}})+p\tilde{w}\end{bmatrix}$$
$$+\frac{\partial}{\partial x}\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ \sigma_{xz}\\ \tilde{u}\sigma_{xx}+\tilde{v}\sigma_{xy}+\tilde{w}\sigma_{xz}+q_x\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ \sigma_{yz}\\ \tilde{u}\sigma_{xy}+\tilde{v}\sigma_{yy}+\tilde{w}\sigma_{yz}+q_y\end{bmatrix} \tag{17}$$
$$+\frac{\partial}{\partial z}\begin{bmatrix}0\\ \sigma_{xz}\\ \sigma_{yz}\\ \sigma_{zz}\\ \tilde{u}\sigma_{xz}+\tilde{v}\sigma_{yz}+\tilde{w}\sigma_{zz}+q_z\end{bmatrix}+\begin{bmatrix}0\\ s_m(\hat{\tilde{u}})\\ s_m(\hat{\tilde{v}})\\ s_m(\hat{\tilde{w}})\\ s_e(\hat{\tilde{u}},\hat{\tilde{v}},\hat{\tilde{w}})\end{bmatrix}=0.$$

The source terms are given by

$$s_m(\cdot)=\rho\left\{\frac{\partial\cdot}{\partial t}+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\cdot}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\cdot}{\partial y}+(\tilde{w}+\hat{\tilde{w}})\frac{\partial\cdot}{\partial z}\right\} \tag{18}$$

$$\begin{aligned}s_e(\hat{\tilde{u}},\hat{\tilde{v}},\hat{\tilde{w}}) &= \tilde{u}s_m(\hat{\tilde{u}})+\tilde{v}s_m(\hat{\tilde{v}})+\tilde{w}s_m(\hat{\tilde{w}})+p\left[\frac{\partial\hat{\tilde{u}}}{\partial x}+\frac{\partial\hat{\tilde{v}}}{\partial y}+\frac{\partial\hat{\tilde{w}}}{\partial z}\right]\\ &+ \sigma_{xx}\frac{\partial\hat{\tilde{u}}}{\partial x}+\sigma_{yy}\frac{\partial\hat{\tilde{v}}}{\partial y}+\sigma_{zz}\frac{\partial\hat{\tilde{w}}}{\partial z}+\sigma_{xy}\left[\frac{\partial\hat{\tilde{v}}}{\partial x}+\frac{\partial\hat{\tilde{u}}}{\partial y}\right]+\sigma_{xz}\left[\frac{\partial\hat{\tilde{w}}}{\partial x}+\frac{\partial\hat{\tilde{u}}}{\partial z}\right]+\sigma_{yz}\left[\frac{\partial\hat{\tilde{w}}}{\partial y}+\frac{\partial\hat{\tilde{v}}}{\partial z}\right]\end{aligned} \tag{19}$$

It is noted that the stress tensors, σ, in SVM are calculated based on velocity derivatives for total velocities u, v and w meaning that they include the induced vortex particle velocities for the calculation of viscous fluxes. This is to eliminate the introduction of dissipative source terms arising from separating velocity derivatives.

The requirement for the particle induced velocities to be included in the CFD as grid velocities means that the velocity equation, Eq. (12) must be evaluated at all M nodes of the CFD grid for all N particles. The computational cost of evaluating this equation directly is $\mathcal{O}(NM)$. Furthermore, computing the source terms, Eqs. (18) and (19), is a further $\mathcal{O}(NM)$ operation if evaluated directly. This is computationally the most expensive part of the hybrid solver as the source terms must be computed at every iteration of the dual-timestepping CFD solver. The Fast Multipole Method (FMM)[23] allows the cost of the velocity computation to be reduced to $\mathcal{O}(N)$.

II.C.2. Fast Multipole Method

The FMM has been applied to the computation of the particle-induced velocity computation, Eq. (12), the particle stretching, Eq. (10), and the source terms, Eqs. (18) and (19). First the formulation for the velocities is presented where a distinction between sources and targets is made. Sources are always the particles whilst the targets can be the particles, if the VPM particle-induced velocities, $\hat{\tilde{\mathbf{u}}}_p$ are being calculated or the CFD grid nodes, if the CFD particle-induced velocities, $\hat{\tilde{\mathbf{u}}}$ are being calculated.

Cheng *et al.*[23] show that Green's function for the Laplace kernel is approximated by the expansions

$$\sum_{i=1}^{N}G\approx\frac{1}{4\pi}\sum_{n=0}^{p}\sum_{m=-n}^{n}\underbrace{r_i^{-n-1}\mathbf{Y}_n^m(\theta_i,\phi_i)}_{S_i}\left\{\sum_{j=1}^{N}\underbrace{\rho_j^n\mathbf{Y}_n^{-m}(\alpha_j,\beta_j)}_{M_j}\right\} \tag{20}$$

and

$$\sum_{i=1}^{N} G \approx \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \underbrace{r_i^n \mathbf{Y}_n^m(\theta_i, \phi_i)}_{R_i} \left\{ \sum_{j=1}^{N} \underbrace{\rho_j^{-n-1} \mathbf{Y}_n^{-m}(\alpha_j, \beta_j)}_{L_j} \right\} \tag{21}$$

Equation (20) corresponds to the multipole expansion and Eq. (21) is the local expansion. In these equations p is the truncation parameter, (r_i, θ_i, ϕ_i) represents position of target i with respect to the centre of expansion in spherical coordinates and $(\rho_j, \alpha_j, \beta_j)$ is the position of source j with respect to the centre of expansion. $\mathbf{Y}_n^m(\alpha, \beta)$ are spherical harmonics which are given by

$$\mathbf{Y}_n^m(\alpha, \beta) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(cos(\alpha)) \exp(im\beta) \tag{22}$$

where $P_n^m(x)$ are the Legendre functions which can be calculated using the recurrence relations

$$\begin{aligned} (n-m)P_n^m(x) &= x(2n-1)P_{n-1}^m(x) - (n+m-1)P_{n-2}^m(x) \\ P_m^m(x) &= (-1)^m(2m-1)!!(1-x^2)^{m/2} \\ P_{m+1}^m(x) &= x(2m+1)P_m^m(x) \end{aligned} \tag{23}$$

Substituting Eqs (20) and (21) into Eq. (12), gives the velocity as

$$\mathbf{u}(\mathbf{x}, t) \approx \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \left\{ \sum_{j=1}^{N} \mathbf{\Gamma}_j M_j \right\} \times \nabla S_i \tag{24}$$

$$\mathbf{u}(\mathbf{x}, t) \approx \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \left\{ \sum_{j=1}^{N} \mathbf{\Gamma}_j L_j \right\} \times \nabla R_i \tag{25}$$

The terms ∇S_i and ∇R_i are calculated by differentiating S_i and R_i with respect to the spherical coordinates (r_i, θ_i, ϕ_i), respectively. The derivative for R_i is given in the paper by Yokota *et al.*,[24] and the computation of ∇S_i follows similarly.

The version of the FMM used in this work begins with a decomposition of the spatial domain into an adaptive octree and follows the algorithm for the adaptive FMM presented in Cheng *et al.*[23]

Following this process, the stretching term, Eq. (10), can be calculated from

$$\frac{d}{dt}\mathbf{\Gamma}(t) \approx \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \left\{ \sum_{j=1}^{N} \mathbf{\Gamma}_j \times \nabla M_j \right\} (\mathbf{\Gamma}_i \cdot \nabla S_i) \tag{26}$$

$$\frac{d}{dt}\mathbf{\Gamma}(t) \approx \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} \left\{ \sum_{j=1}^{N} \mathbf{\Gamma}_j \times \nabla L_j \right\} (\mathbf{\Gamma}_i \cdot \nabla R_i) \tag{27}$$

The source terms given by Eq. (18) can be calculated by the chain rule so that, for example, $s_m(\hat{\tilde{u}})$ is computed from

$$\begin{aligned} \rho \left\{ \frac{\partial \hat{\tilde{u}}}{\partial t} + (\tilde{u} + \hat{\tilde{u}}) \frac{\partial \hat{\tilde{u}}}{\partial x} + (\tilde{v} + \hat{\tilde{v}}) \frac{\partial \hat{\tilde{u}}}{\partial y} + (\tilde{w} + \hat{\tilde{w}}) \frac{\partial \hat{\tilde{u}}}{\partial z} \right\} &= \rho \left\{ \frac{\partial \hat{\tilde{u}}}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \hat{\tilde{u}}}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \hat{\tilde{u}}}{\partial z} \frac{\partial z}{\partial t} + \frac{\partial \hat{\tilde{u}}}{\partial \mathbf{\Gamma}} \frac{\partial \mathbf{\Gamma}}{\partial t} \right. \\ &\quad \left. + (\tilde{u} + \hat{\tilde{u}}) \frac{\partial \hat{\tilde{u}}}{\partial x} + (\tilde{v} + \hat{\tilde{v}}) \frac{\partial \hat{\tilde{u}}}{\partial y} + (\tilde{w} + \hat{\tilde{w}}) \frac{\partial \hat{\tilde{u}}}{\partial z} \right\} \end{aligned} \tag{28}$$

Noting that

$$\frac{\partial x}{\partial t} = \hat{\tilde{u}}_p \qquad \frac{\partial y}{\partial t} = \hat{\tilde{v}}_p \qquad \frac{\partial z}{\partial t} = \hat{\tilde{w}}_p \tag{29}$$

gives the source term as

$$\rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial t}+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}+(\tilde{w}+\hat{\tilde{w}})\frac{\partial\hat{\tilde{u}}}{\partial z}\right\} = \rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial x}\hat{\tilde{u}}_p+\frac{\partial\hat{\tilde{u}}}{\partial y}\hat{\tilde{v}}_p+\frac{\partial\hat{\tilde{u}}}{\partial z}\hat{\tilde{w}}_p+\frac{\partial\hat{\tilde{u}}}{\partial\mathbf{\Gamma}}\frac{\partial\mathbf{\Gamma}}{\partial t}+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}+(\tilde{w}+\hat{\tilde{w}})\frac{\partial\hat{\tilde{u}}}{\partial z}\right\} \tag{30}$$

The computation of $\frac{\partial\hat{\tilde{u}}}{\partial\mathbf{\Gamma}}\frac{\partial\mathbf{\Gamma}}{\partial t}$ is simplified by utilising the fact that the CFD uses a second-order backward difference dual-timestepping scheme. This means that $\frac{\partial\mathbf{\Gamma}}{\partial t}$ can be formulated as

$$\frac{\partial\mathbf{\Gamma}}{\partial t}=\frac{3\mathbf{\Gamma}^{n+1}}{2\Delta t}-\frac{4\mathbf{\Gamma}^{n}}{2\Delta t}+\frac{\mathbf{\Gamma}^{n-1}}{2\Delta t} \tag{31}$$

This results in

$$\frac{\partial\hat{\tilde{u}}}{\partial\mathbf{\Gamma}}\frac{\partial\mathbf{\Gamma}}{\partial t}=\hat{\tilde{u}}^{n+1}\frac{3}{2\Delta t}-\hat{\tilde{u}}^{n}\frac{4}{2\Delta t}+\hat{\tilde{u}}^{n-1}\frac{1}{2\Delta t} \tag{32}$$

and $\hat{\tilde{u}}^{n+1}$, $\hat{\tilde{u}}^{n}$ and $\hat{\tilde{u}}^{n-1}$ can be calculated using the FMM and Eqs. (24) and (25), by substituting $\mathbf{\Gamma}^{n+1}$, $\mathbf{\Gamma}^{n}$ and $\mathbf{\Gamma}^{n-1}$ for $\mathbf{\Gamma}$, respectively. Finally the $\frac{\partial\hat{\tilde{\mathbf{u}}}}{\partial\mathbf{x}}$ terms are calculated using

$$\frac{\partial\hat{\tilde{\mathbf{u}}}}{\partial\mathbf{x}}\approx\frac{1}{4\pi}\sum_{n=0}^{p}\sum_{m=-n}^{n}\left\{\sum_{j=1}^{N}\mathbf{\Gamma}_j\times\nabla M_j^T\right\}\times\nabla S_i \tag{33}$$

$$\frac{\partial\hat{\tilde{\mathbf{u}}}}{\partial\mathbf{x}}\approx\frac{1}{4\pi}\sum_{n=0}^{p}\sum_{m=-n}^{n}\left\{\sum_{j=1}^{N}\mathbf{\Gamma}_j\times\nabla L_j^T\right\}\times\nabla R_i \tag{34}$$

So the source term given by Eq. (28) is calculated by computing Eq. (32) and Eqs. (33) and (34) and then substituting these into Eq. (28). The source terms involving the other induced velocity components are calculated in a similar way.

II.C.3. Particle seeding

Vortex particles are initialised or "seeded" according to the vorticity in the CFD solution. A region in space, either a line in 2D or a plane in 3D, is identified and if the magnitude of vorticity at any point in this region is larger than a user-specified threshold, particles are seeded. Using the 2D case, shown in Figure 2, as the example, the user specifies the length of the line and the position along the x-axis. The particle spacing, h is set to

$$h=u_{ref}\Delta t \tag{35}$$

so that no vorticity from the CFD is missed. The number of particles in the y-direction, N_y, is set by

$$N_y=\frac{y_{max}-y_{min}}{h} \tag{36}$$

so that each particle lies in squares of equal area, h^2. In three-dimensions the number of particles in the z-direction is also set in a similar manner so that each particle lies in a cube of equal volume, h^3. The particle radius, σ, is then calculated by

$$\sigma=\frac{h}{\beta} \tag{37}$$

where β is the particle overlap ratio, which should be less than 1 to ensure that particles overlap.[25] The vorticity at the centre of each particle is interpolated from the vorticity at the CFD nodes using an inverse distance weighting so that for particle i the vorticity is given by

$$\omega_i(\mathbf{x})=\begin{cases}\dfrac{\sum_{j=1}^{N}s_j(\mathbf{x})\omega_j}{\sum_{j=1}^{N}s_j(\mathbf{x})} & \text{if } d(\mathbf{x},\mathbf{x}_j)\neq 0 \text{ for all } j\\ \omega_j & \text{if } d(\mathbf{x},\mathbf{x}_j)=0 \text{ for some } j\end{cases} \tag{38}$$

In this equation

$$s_j(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x}_j)^p} \tag{39}$$

where d is the distance metric and p is the power parameter. Using a larger value of p has the effect of increasing the influence of solution from the closest CFD points. In this work, a value of $p = 7$ was used as smaller values resulted in highly unsmooth vorticity distributions, which eventually results in the method breaking. Now each particle has an associated vorticity the strength of the particles that gives this vorticity distribution must be found. In almost all the literature the particle strengths are initialised by multiplying the particle vorticity by the volume of the particle, i.e.

$$\Gamma_i = \omega_i h^d, \tag{40}$$

where d is the dimension of the problem. This introduces an error, known as a regularisation or smoothing error,[26] which for most Vortex Particle Methods that appear in the literature, appears to be acceptable. However, in this work, this means that the core vorticity is not captured entirely and this has the effect of dissipating the vortex. An alternative is to solve Eq. (4), which can be written as

$$\mathbf{A\Gamma} = \omega, \tag{41}$$

where

$$A_{ij} = \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i). \tag{42}$$

The condition number of the matrix $\mathbf{A}$ is dependent on the value of σ, which itself is dependent on the inter-particle spacing and the overlap ratio. The condition number of the matrix correlates with σ, so in order to achieve a better conditioned system with the inter-particle spacing can be reduced by increasing the number of particles or the overlap ratio can be increased. Unfortunately, the VPM requires the overlap ratio to be less than one to ensure convergence[25] and computational resources restrict the number of particles that can be seeded on each timestep, especially when simulating long times. Therefore, a way of solving this ill-conditioned system is needed. This system is solvable using Singular Value Decomposition (SVD) but results in an odd-even coupling of the particle strengths. This then causes numerical errors when the velocities are computed. Therefore, the problem became one of solving Eq. (41) for smooth particle strengths. This was achieved by adding the fourth order part of Jameson dissipation to the equation to damp out the high frequencies.

III. Results

First, results are presented for a two-dimensional test to demonstrate the applicability of this method. Results are then presented for a three-dimensional test case.

III.A. Two-dimensional test case

The two-dimensional test case involved the interaction of an artificially generated vortex with a NACA 0021 aerofoil. The test was carried out on two different meshes. Both meshes had the same point distribution in the aerofoil near field but one of the meshes, the coarse mesh shown in Figure 3, coarsened away from this point whilst the other, the fine mesh shown in Figure 4, maintained this node density along the vortex path. The fine mesh has a total of 139,810 points whilst the coarse mesh consists of 19,012 points.

The hybrid CFD-VPM approach was only applied to the coarse mesh but this aerofoil-vortex interaction test case was simulated on both meshes using the CFD solver on its own. The vortex was initially inserted into the CFD with its centre located directly above the aerofoil trailing edge. All the simulations were performed using a timestep size of 1.0×10^{-4} seconds and were run for 150 timesteps.

For the hybrid solver, the particle seeding line was placed at 30% chord downstream of the trailing edge and was sized vertically so it was slightly larger than the vortex. The hybrid solver was set up to attempt to seed particles, with an overlap ratio of 0.5, at every time step. The result was that 22 particles were seeded every timestep until the 56th timestep by which point the vortex had passed through the seeding line completely. This meant that the computation reached a maximum of 1210 particles.

Figures 5, 6 and 7 show the vorticity distribution at four different timesteps for the CFD only computation on the fine mesh, CFD-only computation on the coarse mesh and hybrid computation on the coarse mesh

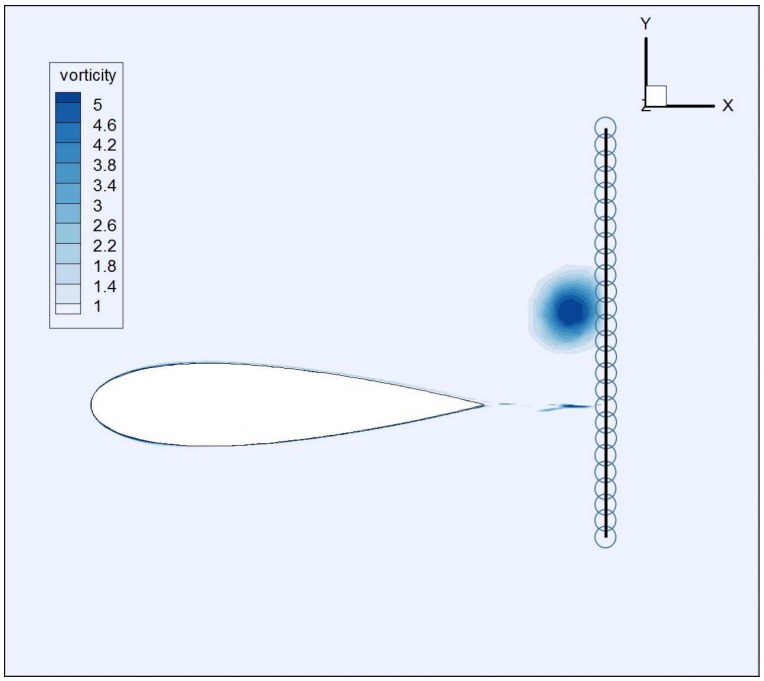


Figure 2. Example of particles being seeded in 2D.

respectively. As can be seen the coarse mesh is highly dissipative for the CFD-only computation but by coupling a VPM to the CFD solver the vortex is preserved. Furthermore, the CFD-only simulation on the fine mesh took 35% longer than the simulation using the hybrid solver meaning the hybrid solver is computationally more efficient. The results shown here for the hybrid simulation were produced by summing the vorticity produced by the CFD with the vorticity produced by the vortex particles and then interpolating this onto the fine mesh. The particles at each of these timesteps can be seen in Figure 8. This shows how the number of particles changes at each time and how they wrap up to the shape of the vortex as the simulation progresses.

III.B. Three-dimensional test case

The hybrid solver has also been tested on a three-dimensional test case. The simulation was set up to correspond to the experimental investigations carried out by McAlister and Takahashi.[27] One of their experiments involved a half model of a NACA 0015 wing with an aspect ratio of 6.6, unit chord and square tip, positioned at an angle of attack of 12°. The freestream conditions correspond to a Reynolds number of 1.5×10^6 and Mach number of 0.13. An unstructured mesh was generated using SOLAR,[28] which consisted of 360,909 grid nodes. This constitutes the coarse mesh for the 3D test case. This mesh was then run using the adaptation routine supplied in TAU in order to generate a finer mesh in the region of the trailing tip vortex only. The final mesh generated by this procedure consisted of 3,819,630 grid nodes. The coarse mesh was run using the CFD solver alone and the coupled CFD-VPM solver whilst the fine mesh was run using the CFD solver alone only.

When run with the CFD solver alone the coarse mesh results in diffusion of the vortex as can be seen in Figure 9, where the core strength has decreased at a location just 50% chord downstream of the trailing edge, Figure 9(b), and by 4 chord lengths downstream of the trailing edge, Figure 9(d), there is no evidence of a vortex at all. The simulation was run using a timestep size of 1×10^{-3} seconds and all the results presented in this section are at the 100th timestep. Comparing these results to those produced using the fine mesh, Figure 10, emphasises the amount of dissipation present in the CFD due to inadequate spatial resolution. Running this same case with the hybrid solver on the coarse mesh, results in the preservation of the vortex structure as seen in Figure 11. This hybrid solver was set up to attempt to seed particles at every timestep and the particle overlap ratio was set to 0.6. The seeding panel was positioned 30% chord downstream of

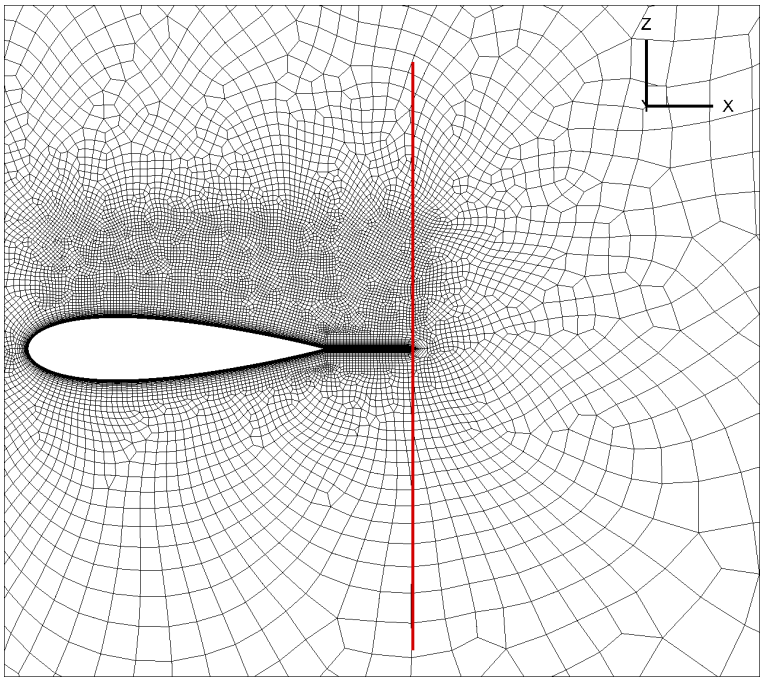


Figure 3. Coarse mesh used in NACA 0021 2D test case consisting of 19012 points

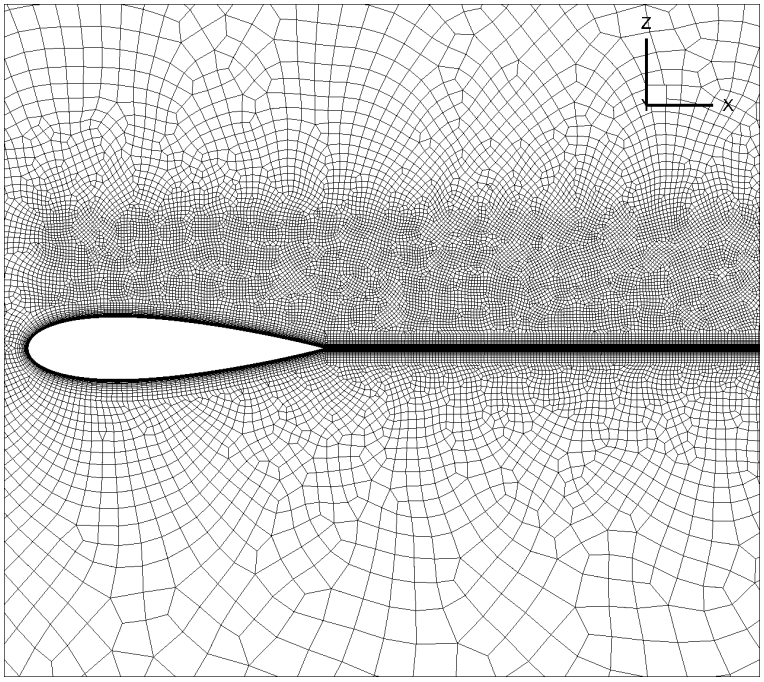
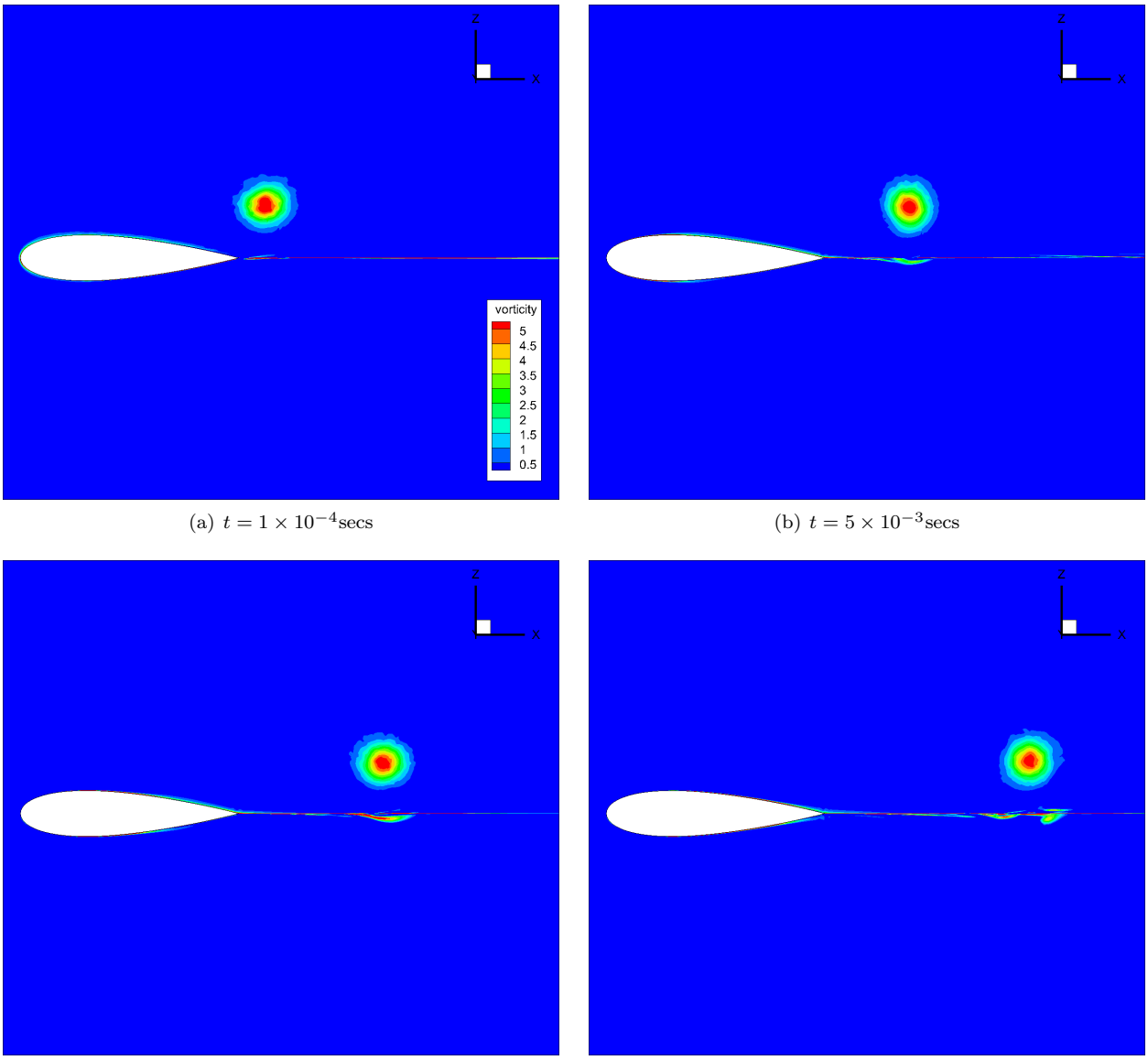


Figure 4. Fine mesh used in NACA 0021 2D test case consisting of 139810 points

the trailing edge and by the 100th timestep there were 11,979 particles. The results shown in Figure 11 have been produced by evaluating the vorticity induced by the particles on the fine mesh and summing this with the solution produced by the CFD part of the hybrid solver after it has been interpolated onto the fine mesh. This method of producing the results is purely for visualisation purposes. Comparison of the results of the coupled CFD-VPM solver on the coarse mesh with those of the CFD-only solver on the fine mesh



(a) $t = 1 \times 10^{-4}$secs

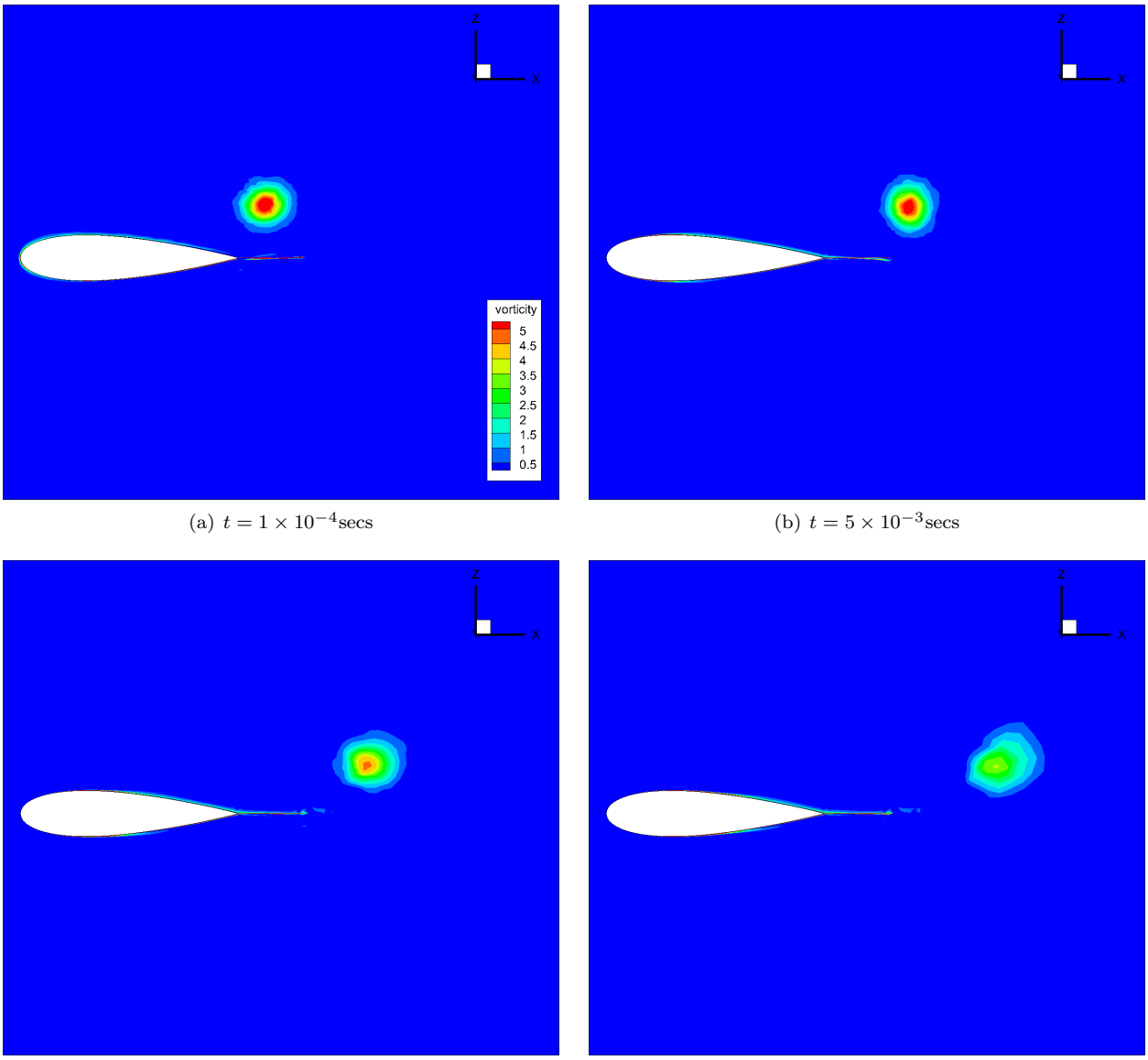
(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Figure 5. Results of the 2D tip vortex simulation using the CFD solver only on the fine mesh

shows that the vortex is captured fairly well up to 50% downstream of the trailing edge, Figures 11(b) and 10(b), but by 1 chord length downstream of the trailing edge the core of the vortex from the hybrid solver is not as intense as the fine mesh, Figures 11(c) and 10(c). There are a couple of possible reasons for this. The first is that the vortex has already dissipated slightly by the time it is converted into particles at 30% downstream of the trailing edge. The second is that there are not enough particles to sufficiently capture the core. As previously stated the size of the timestep dictates the size of the particles, which gives the number of particles. This means, with the timestep size used in this work, there were 121 particles seeded on each timestep, which means there are 11 particles across the box in the y- and z-directions. The size of the seeding panel relative to the size of the vortex is shown in Figure 12. This would suggest that currently the inner core is covered by only a few particles. Despite the core not being captured completely the general structure of the vortex does appear to be captured quite well as shown in Figure 13. Here the iso-surface of vorticity at a value of 0.6 is plotted and shows that the coupled CFD-VPM solver results agree well with the fine mesh results. Finally, Figure 14 shows the position of the particles on the 100th timestep. The particles behave as expected, twisting around in an anti-clockwise direction as they convect downstream.



(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

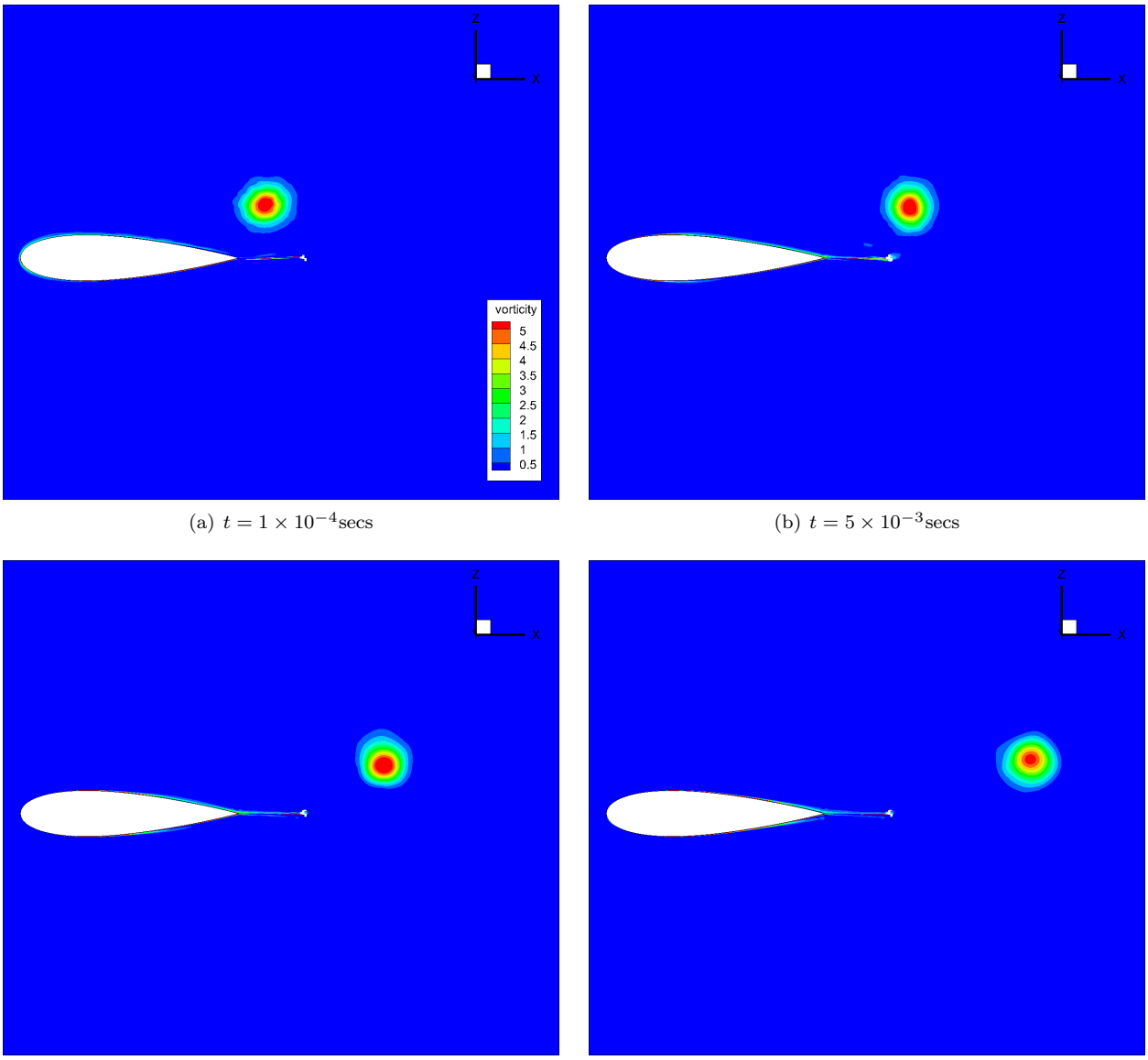
Figure 6. Results of the 2D tip vortex simulation using the CFD solver only on the coarse mesh

IV. Conclusion

A coupled CFD-VPM solver has been created. It has been shown to preserve the vortex structure on a coarse CFD mesh whilst also being computationally more efficient compared to running the same simulation using the CFD solver alone on a fine mesh in both two and three dimensions. The computational cost of the method has been reduced by employing a Fast Multipole Method to compute the induced velocities and source terms necessary for the Split Velocity Method, which is used to couple the two solvers. The next steps in this work would be to introduce a particle merging scheme and parallelize the code to enable more realistic and meaningful simulations to be performed in an adequate timeframe.

Acknowledgments

The research leading to these results was co-funded by Innovate UK, the UKs innovation agency, within the Enhanced Fidelity Transonic Wing project.



(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Figure 7. Results of the 2D tip vortex simulation using the coupled CFD-VPM solver on the coarse mesh

References

[1]Srinivasan, G. R., "Computations of Two-Dimensional Airfoil-Vortex Interactions," Tech. Rep. 3885, NASA, 1985.

[2]Potsdam, M., Yeo, H., and Johnson, W., "Rotor airloads prediction using loose Aerodynamic/Structural coupling," *American Helicopter Society 60th Annual Forum*, 2004.

[3]Abate, M., "Gust boundary condition for the DLR TAU-Code," Tech. rep., DLR, 2007.

[4]Rodriguez, I., Lehmkuhl, O., Borrell, R., and Oliva, A., "Direct numerical simulation of a NACA0012 in full stall," *International Journal of Heat and Fluid Flow*, Vol. 43, October 2013, pp. 194–203.

[5]Svard, M., Lundberg, J., and Nordström, J., "A computational study of vortex-airfoil interaction using high-order finite difference methods," *Computers and Fluids*, Vol. 39, No. 8, 2010, pp. 1267–1274.

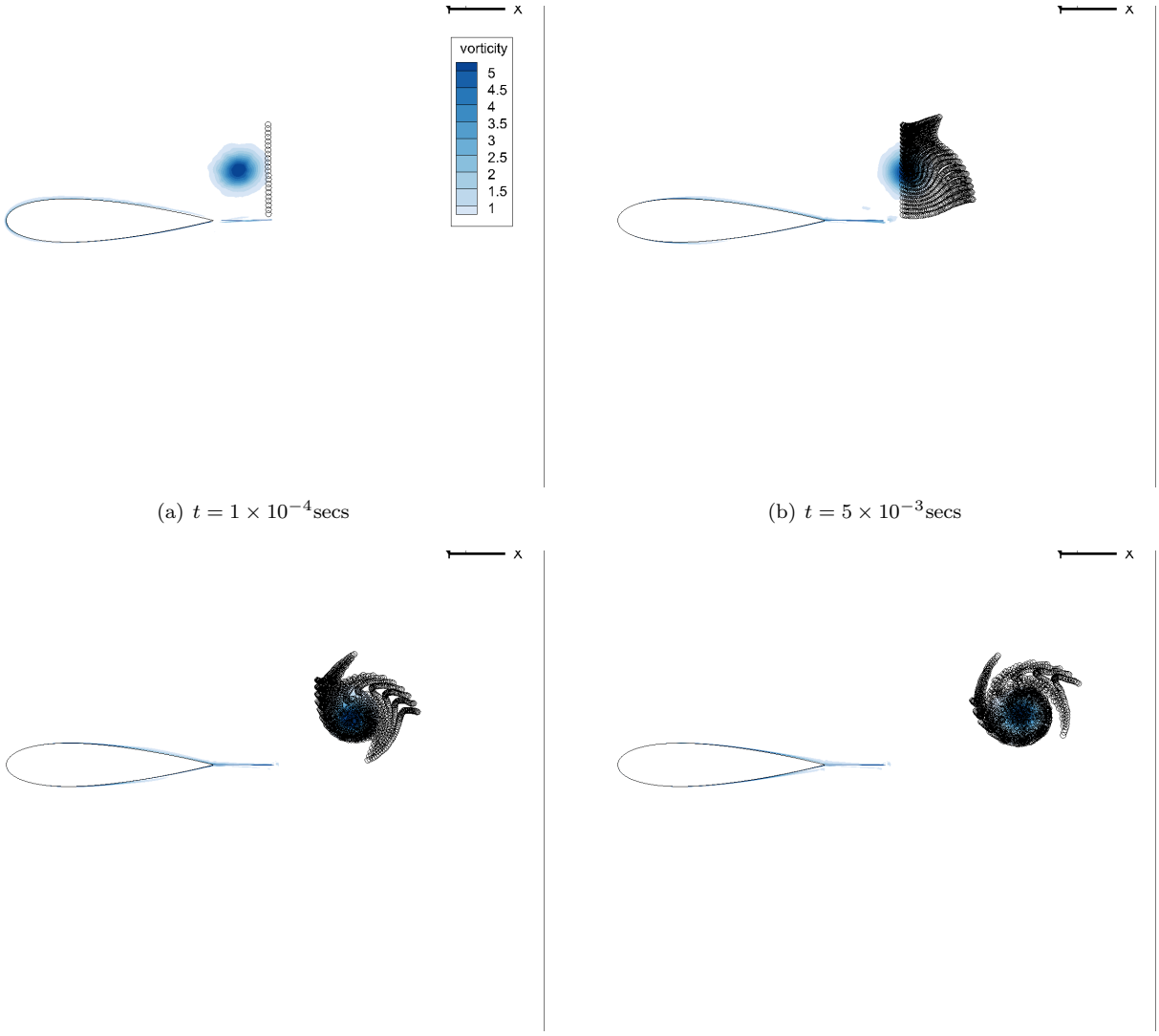
[6]Wolf, C., *A Chimera Simulation method and Detatched Eddy Simulation for Vortex-Airfoil Interactions*, Ph.D. thesis, University of Göttingen, 2010.

[7]Cottet, G. and Koumoutsakos, P. D., *Vortex Methods: Theory and Practice*, Cambridge Univ Pr, 2000.

[8]Rossow, V., "Inviscid modeling of aircraft trailing vortices," Tech. Rep. 409, NASA, 1977.

[9]Smith, S. and Kroo, I., "Induced Drag Computations on Wings with Accurately MoModel Wakes," *Journal of Aircraft*, Vol. 34, No. 2, 2006, pp. 253–255.

[10]Chatelain, P., Curioni, A., Bergdorf, M., Rossinelli, D., Andreoni, W., and Koumoutsakos, P., "Billion vortex particle direct numerical simulation of aircraft wakes," *Comput. Methods Appl. Mech. Engrg*, Vol. 197, No. 13-16, 2008, pp. 1296–1304.



(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Figure 8. Particle visualisation as the simulation progresses

[11]Röttgermann, A., Behr, R., Schöttl, C., and Wagner, S., *Calculation of Blade-Vortex Interaction of Rotary Wings in Incompressible Flow by an Unsteady Vortex-Lattice Method Including Free Wake Analysis*, Vieweg+Teubner Verlag, Wiesbaden, 1992, pp. 153–166.

[12]Padakannaya, R., "The Vortex Lattice Method for the Rotor-Vortex Interaction Problem," Tech. Rep. 2421, NASA, 1974.

[13]Sitaraman, J., *CFD based unsteady aerodynamic modeling for rotor aeroelastic analysis*, Ph.D. thesis, University of Maryland, 2003.

[14]Anusonti-Inthra, P. and Floros, M., "Coupled CFD and Particle Vortex Transport Method: Wing Performance and Wake Validations," *38th Fluid Dynamics Conference and Exhibit*, 2008.

[15]Zhao, j., He, C., Zhang, L., Zhao, H., and Hu, P., "Coupled Viscous Vortex Particle Method and Unstructured Computational Fluid Dynamics Solver for Rotorcraft Aerodynamic Interaction Analysis," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

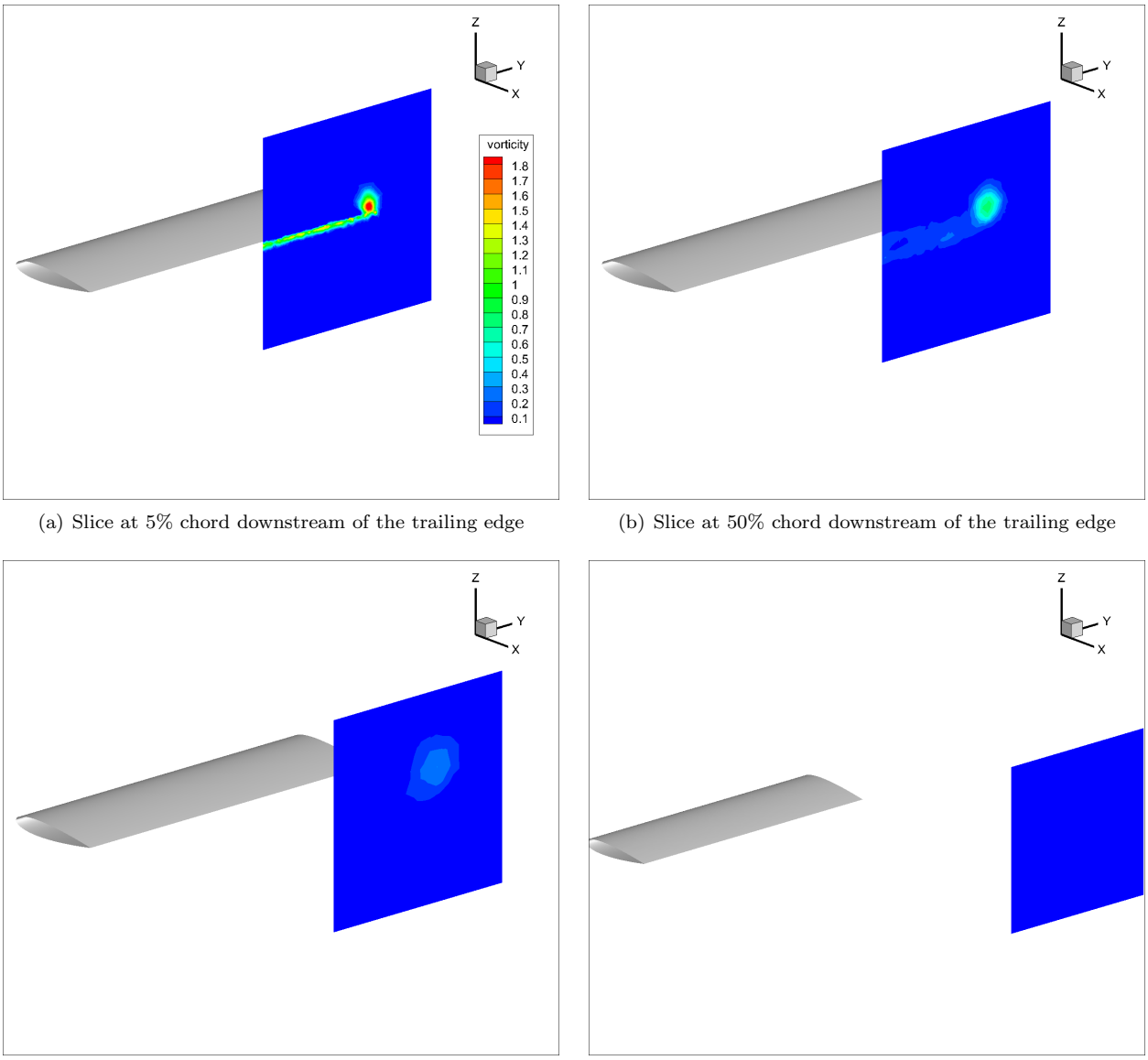
[16]Stone, C., Duque, E., Hennes, C., and Gharakhani, A., "Rotor wake modeling with a coupled Eulerian and Vortex Particle Method," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

[17]Stock, M., Gharakhani, A., and Stone, C., "Modeling Rotor Wakes with a Hybrid OVERFLOW-Vortex Method on a GPU Cluster," *28th AIAA Applied Aerodynamics Conference*, 2010.

[18]Pahla, A., Manickathan, L., Ferreira, C., and Van Bussel, G., "A hybrid eulerian-lagrangian flow solver," arXiv preprint arXiv:1505.03368.

[19]Wales, C., Gaitonde, A., and Jones, D., "Simulation of airfoil gust response using prescribed velocities," *IFASD*, 2011.

[20]Huntley, S., Jones, D., and Gaitonde, A., "2D and 3D gust repsonse using a prescribed velocity method," *46th AIAA Fluid Dynamics Conference*, 2016.



(a) Slice at 5% chord downstream of the trailing edge

(b) Slice at 50% chord downstream of the trailing edge

(c) Slice at 1 chord downstream of the trailing edge

(d) Slice at 4 chords downstream of the trailing edge

Figure 9. Results of the 3D wingtip vortex simulation using the CFD solver only on coarse mesh

[21]Speck, R., *Generalized Algebraic Kernels and Multipole Expansions for Massively Parallel Vortex Particle Methods*, Ph.D. thesis, Universität Wuppertal, 2011.

[22]Winckelmans, G. and Leonard, A., "Contributions to Vortex Particle Methods for the computation of three-dimensional incompressible unsteady flows," *Journal of Computational Physics*, Vol. 109, 1993, pp. 247–273.

[23]Cheng, H., Greengard, L., and Rokhlin, V., "A fast adaptive multipole algorithm in three dimensions," *Journal of Computational Physics*, Vol. 155, 1999, pp. 466–498.

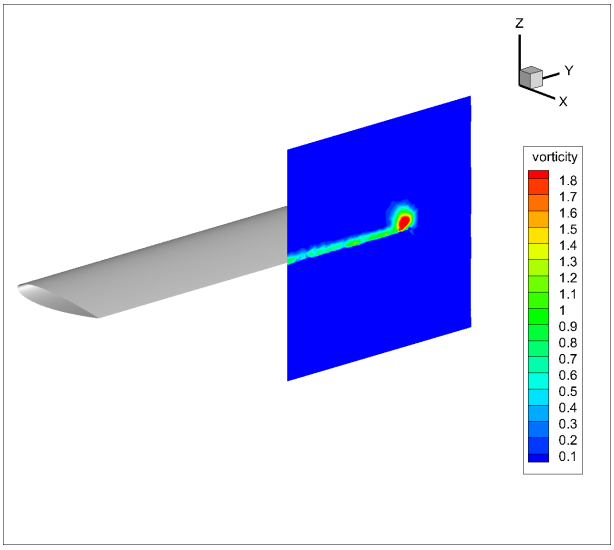
[24]Yokota, R., Narumi, T., Sakamaki, R., Kameoka, S., Obi, S., and Yasuoka, K., "Fast multipole methods on a cluster of GPUs for the meshless simulation of turbulence," *Computer Physics Communications*, Vol. 180, 2009, pp. 2066–2078.

[25]Barba, L., Leonard, A., and Allen, C., "Advances in viscous vortex methods-meshless spatial adaptation based on radial basis function interpolation," *Int. J. Numer. Meth. Fluids*, Vol. 47, 2005, pp. 387–421.

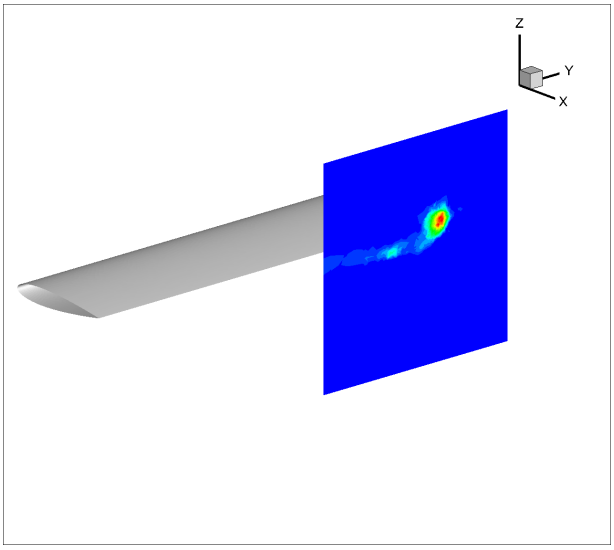
[26]Barba, L. and Rossi, L., "Global field interpolation for particle methods," *Journal of Computational Physics*, Vol. 229, 2010, pp. 1292–1310.

[27]McAlister, K. and Takahashi, R., "NACA0015 Wing Pressure and Trailing Vortex Measurements," Tech. Rep. 3151, NASA, 1991.

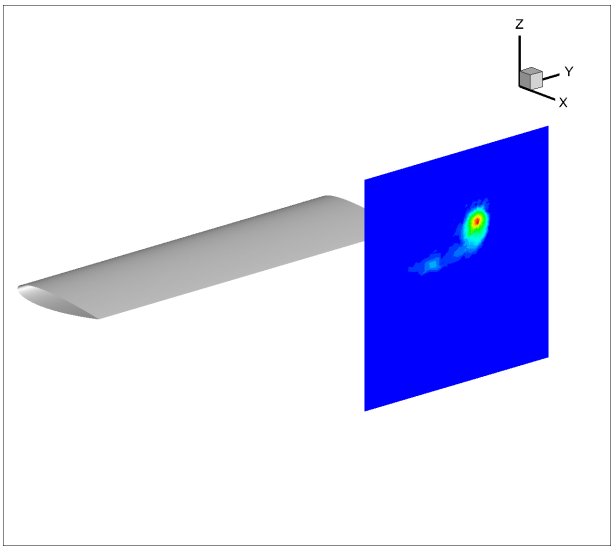
[28]Martineau, D., Stokes, S., Munday, S., Jackson, A., Gribben, B., and Verhoeven, N., "Anisotropic hybrid mesh generation for industrial RANS applications," *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.



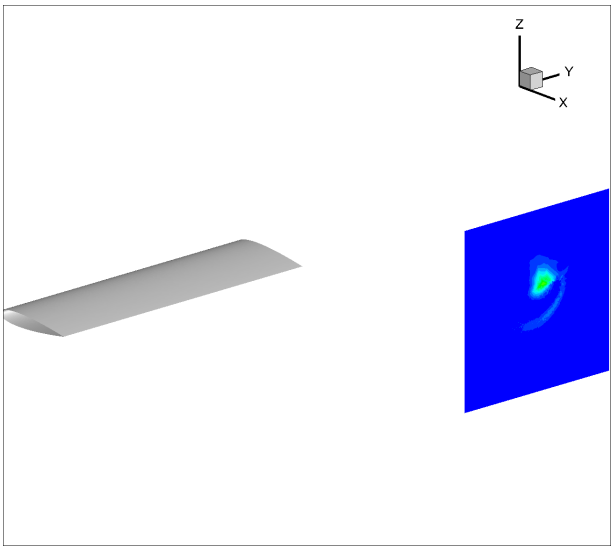
(a) Slice at 5% chord downstream of the trailing edge



(b) Slice at 50% chord downstream of the trailing edge

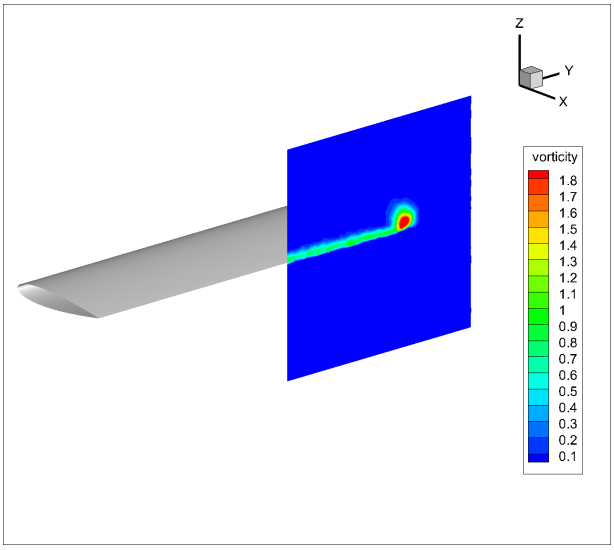


(c) Slice at 1 chord downstream of the trailing edge

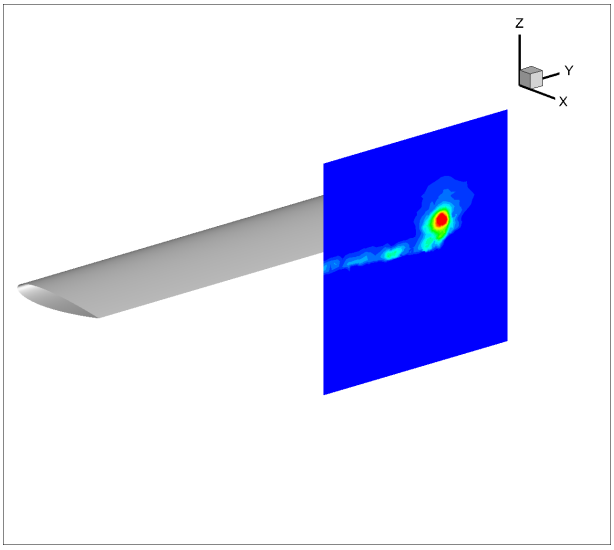


(d) Slice at 4 chords downstream of the trailing edge

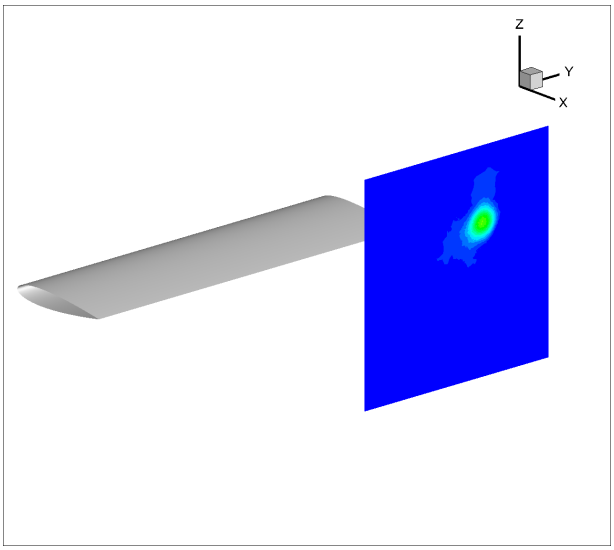
Figure 10. Results of the 3D wingtip vortex simulation using the CFD solver only on a fine mesh



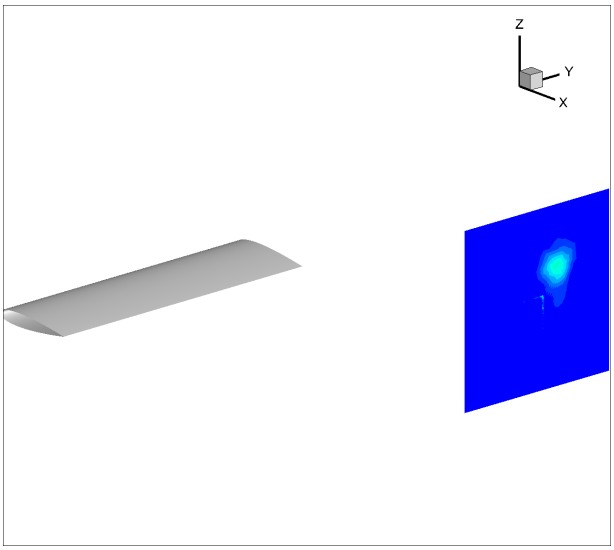
(a) Slice at 5% chord downstream of the trailing edge



(b) Slice at 50% chord downstream of the trailing edge



(c) Slice at 1 chord downstream of the trailing edge



(d) Slice at 4 chords downstream of the trailing edge

Figure 11. Results of the 3D wingtip vortex simulation using the coupled CFD-VPM solver on the coarse mesh

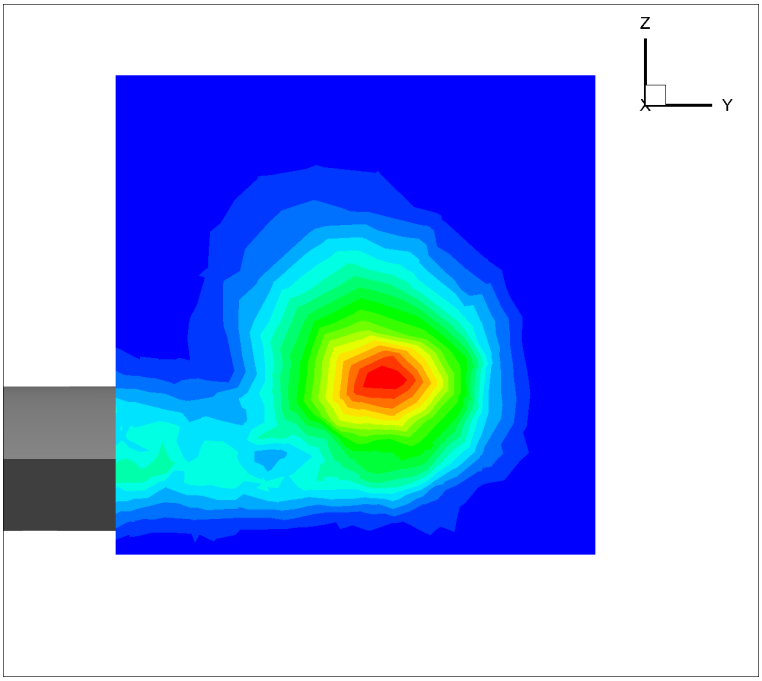
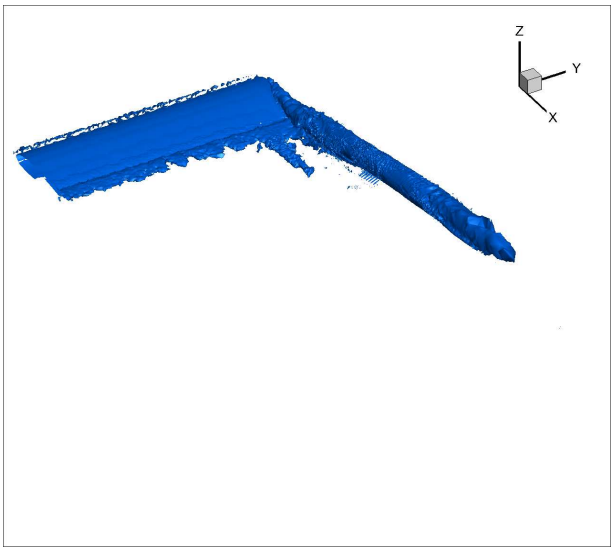
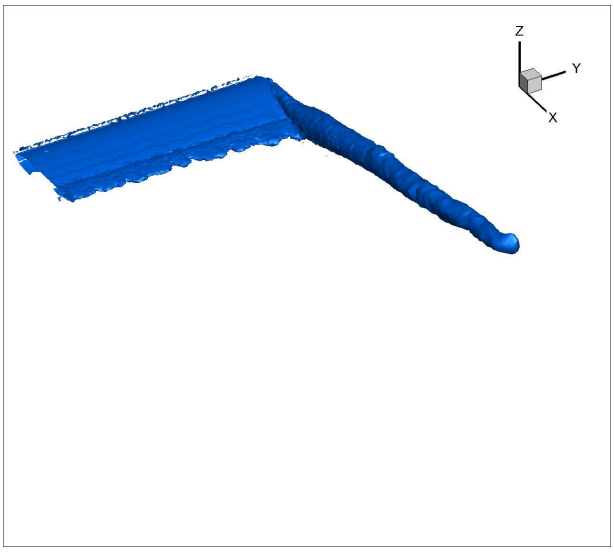


Figure 12. Size of the seeding panel used in the coupled CFD-VPM solver



(a) Vorticity iso-surface for CFD-only solver on the fine mesh



(b) Vorticity iso-surface for coupled CFD-VPM solver on the coarse mesh

Figure 13. Iso-surface of vorticity, $\omega = 0.6$, of the 3D wingtip vortex simulation

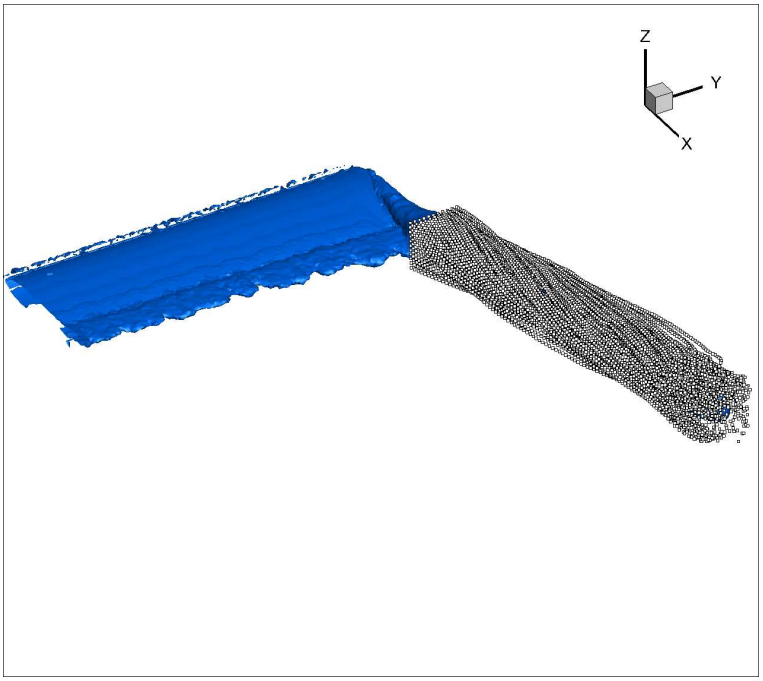


Figure 14. Location of the particles at the 100th timestep of the coupled CFD-VPM solver